

# STUDY AND IMPLEMENTATION OF 3G MOBILE SECURITY

Thesis submitted in partial fulfillment of the requirements for the award of the degree of

**Bachelor of Technology in Computer Science and Engineering**

by:

Sutirtha Prakash (10606019)

Sachikanta Behera (10606029)



Department of Computer Science and Engineering

National Institute of Technology

ROURKELA

# STUDY AND IMPLEMENTATION OF 3G MOBILE SECURITY

Thesis submitted in partial fulfillment of the requirements for the award of the degree of

**Bachelor of Technology in Computer Science and Engineering**

by:

Sutirtha Prakash (10606019)

Sachikanta Behera (10606029)

Under guidance of

Prof. A. K. Turuk



Department of Computer Science and Engineering

National Institute of Technology

ROURKELA



National Institute of Technology

Rourkela

## CERTIFICATE

This is to certify that the work in this Thesis Report entitled “3G Mobile Security” submitted by Sutirtha Prakash (10606019) and Sachikanta Behera (10606029), has been carried out under my supervision and guidance, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science during session 2006-2010 in the Department of Computer Science and Engineering, National Institute of Technology, Rourkela.

This work is to study the 3G network and implementation of security mechanisms in network access security of the 3G security architecture.

Date:

Prof. A.K. Turuk

Department of Computer Science

NIT Rourkela

## ACKNOWLEDGEMENT

No thesis is created entirely by an individual, many people have helped to create this thesis and each of their contribution has been valuable. We express our sincere gratitude to our thesis supervisor, Prof. A. K. Turuk, Department of Computer Science and Engineering, for his kind and able guidance for the completion of the thesis work. His consistent support and intellectual guidance made us energize and innovate new ideas. Last, but not least we would like to thank all the professors and lecturers, and members of the Department of Computer Science and Engineering, National Institute of Technology, Rourkela for their generous help in various ways for the completion of this thesis.

Sutirtha Prakash

10606019

Sachikanta Behera

10606029

## ABSTRACT

In the last decade there has been an exponential rise in use of mobile devices. 3G is the latest mobile technology that is currently in widespread use. The Universal Mobile Telecommunications System (UMTS) is the most popular third generation mobile communication systems, which reposes on the popularity of the ‘second generation’ GSM system by introducing high quality services while retaining its essential and robust security features. Wireless communication is less secure, and mobility entails higher security risks than stationary devices. Security is the foremost concern in today’s mobile communication systems. Latest security mechanisms are needed to protect the singular features introduced in 3G technology. The security framework for 3G mobile networks is considered, and the various protocols for protection of the network access interface are studied and analyzed.

## CONTENTS

SECTION	DESCRIPTION	PAGE NO.
	ABSTRACT	5
CHAPTER 1	INTRODUCTION	9
CHAPTER 2	ARCHITECTURE	10
2.1	OVERVIEW OF 3G ARCHITECTURE	10
2.2	3G SECURITY ARCHITECTURE	11
2.3	UMTS ARCHITECTURE	13
CHAPTER 3	NETWORK ACCESS SECURITY	14
3.1	USER IDENTITY CONFIDENTIALITY	14
3.2	AUTHENTICATION AND KEY AGREEMENT	14
3.3	MILENAGE ALGORITHM	17
3.4	INTEGRITY PROTECTION OF SIGNALING MESSAGE	21
3.5	DATA CONFIDENTIALITY	22
3.6	KASUMI ALGORITHM	24
CHAPTER 4	IMPLEMENTATION DETAILS	30
4.1	AUTHENTICATION AND KEY AGREEMENT	30
4.2	SOCKET PROGRAMMING	31
4.3	CONFIDENTIALITY AND INTEGRITY	32
4.4	SCREENSHOTS	33
CHAPTER 5	CONCLUSION AND FUTURE WORK	35
CHAPTER 6	REFERENCES	36

## LIST OF FIGURES

<b>SL.</b>	<b>NAME OF FIGURE</b>	<b>PAGE NO.</b>
FIG1	3G rel99 architecture	11
FIG2	Overview of UMTS security architecture	13
FIG3	Authentication and Key Agreement procedure	16
FIG4	Definition of f1, f1*, f2, f3, f4, f5 and f5*	20
FIG5	Derivation of MAC(or XMAC) on a signaling message	21
FIG6	Ciphering over radio access link	22
FIG7	Overall Setup of 3G Security	23
FIG8	f8 Key stream Generator	26
FIG9	f9 integrity function	28
FIG10	Screenshot 1	33
FIG11	Screenshot 2	34

## LIST OF ABBREVIATIONS

<b>3GPP</b>	Third Generation Partnership Project
<b>UMTS</b>	Universal Mobile Telecommunications System
<b>AKA</b>	Authentication and key agreement
<b>AMF</b>	Authentication management field
<b>AN</b>	Access Network
<b>AUTN</b>	Authentication Token
<b>AUTS</b>	Re-synchronization token
<b>AuC</b>	Authentication Centre
<b>AV</b>	Authentication Vector
<b>CN</b>	Core Network

<b>CS</b>	Circuit Switched
<b>FRESH</b>	Random value used to prevent replay of signaling messages
<b>GGSN</b>	Gateway GPRS Support Node
<b>GSM</b>	Global System for Mobile communications
<b>HE</b>	Home Environment
<b>HLR</b>	Home Location Register
<b>IMSI</b>	International Mobile Subscriber Identity
<b>MAC</b>	Message Authentication Code
<b>MAC-A</b>	The message authentication code included in AUTN, computed using f1
<b>ME</b>	Mobile Equipment
<b>MAP</b>	Mobile Application Part
<b>MS</b>	Mobile Station
<b>MSC</b>	Mobile Services Switching Centre
<b>OSA</b>	Open Service Architecture
<b>PS</b>	Packet Switched
<b>RAND</b>	Random challenge
<b>RES</b>	(expected) user response to challenge in GSM
<b>RNC</b>	Radio Network Controller
<b>SQN</b>	Sequence number
<b>SQNHE</b>	Sequence number counter maintained in the HLR/AuC
<b>SQNMS</b>	Sequence number counter maintained in the USIM
<b>SGSN</b>	Serving GPRS Support Node
<b>SN</b>	Serving Network
<b>TMSI</b>	Temporary Mobile Subscriber Identity
<b>IMSI</b>	International Mobile Subscriber Identity
<b>VLR</b>	Visitor Location Register
<b>XRES</b>	Expected Response
<b>XMAC</b>	Expected Message Authentication Code
<b>  </b>	Concatenation
<b>⊕</b>	Exclusive or
<b>AK</b>	Anonymity Key used in 3G
<b>CK</b>	Cipher Key used in 3G
<b>f1</b>	Message authentication function used to compute MAC
<b>f2</b>	Message authentication function used to compute RES and XRES
<b>f3</b>	Key generating function used to compute CK
<b>f4</b>	Key generating function used to compute IK
<b>f5</b>	Key generating function used to compute AK
<b>f8</b>	3G ciphering function
<b>f9</b>	3G integrity function
<b>IK</b>	Integrity Key used in 3G
<b>K</b>	Shared secret key used in 3G between the operator and the user



---

## *CHAPTER 1*

---

### INTRODUCTION

In the last decade there has been a proliferation in the use of mobile technology for communication. The rapid growth in use of mobile devices and the advancement of technology led to the introduction of high end and cheap mobile equipments which can support high quality mobile services. The third generation (3G) mobile technology has much superior bandwidth than 2G and supports high quality data and voice services. Universal Mobile Telecommunication System (UMTS), standardized by the 3GPP, is the 3G mobile communication technology successor to GSM and GPRS. UMTS enhances the existing GSM technology by providing increased bandwidth, data capacity and a wide range of high end services and features using a unique radio interface standard known as UMTS Terrestrial Radio Access (UTRA). Apart from normal talking services users can now use interactive services like internet access, chat services, online banking, data transfer, music and movies download etc. But as services increase and mobile networks become more complex and open, so do the security risks and type of attacks from potential hackers. Valuable and precious information sent through wireless networks has to be protected from potential hackers. The complex network configuration, which allow superior connectivity rates and “on the go” connectivity, may increase the probability of possible attacks. In addition, the introduction of IP layer [1] in the network domain, for signaling and user data transmission, makes the network open and more vulnerable. UMTS security architecture as proposed by 3GPP retains and enhances the essential features of GSM security.

---

## *CHAPTER 2: ARCHITECTURE*

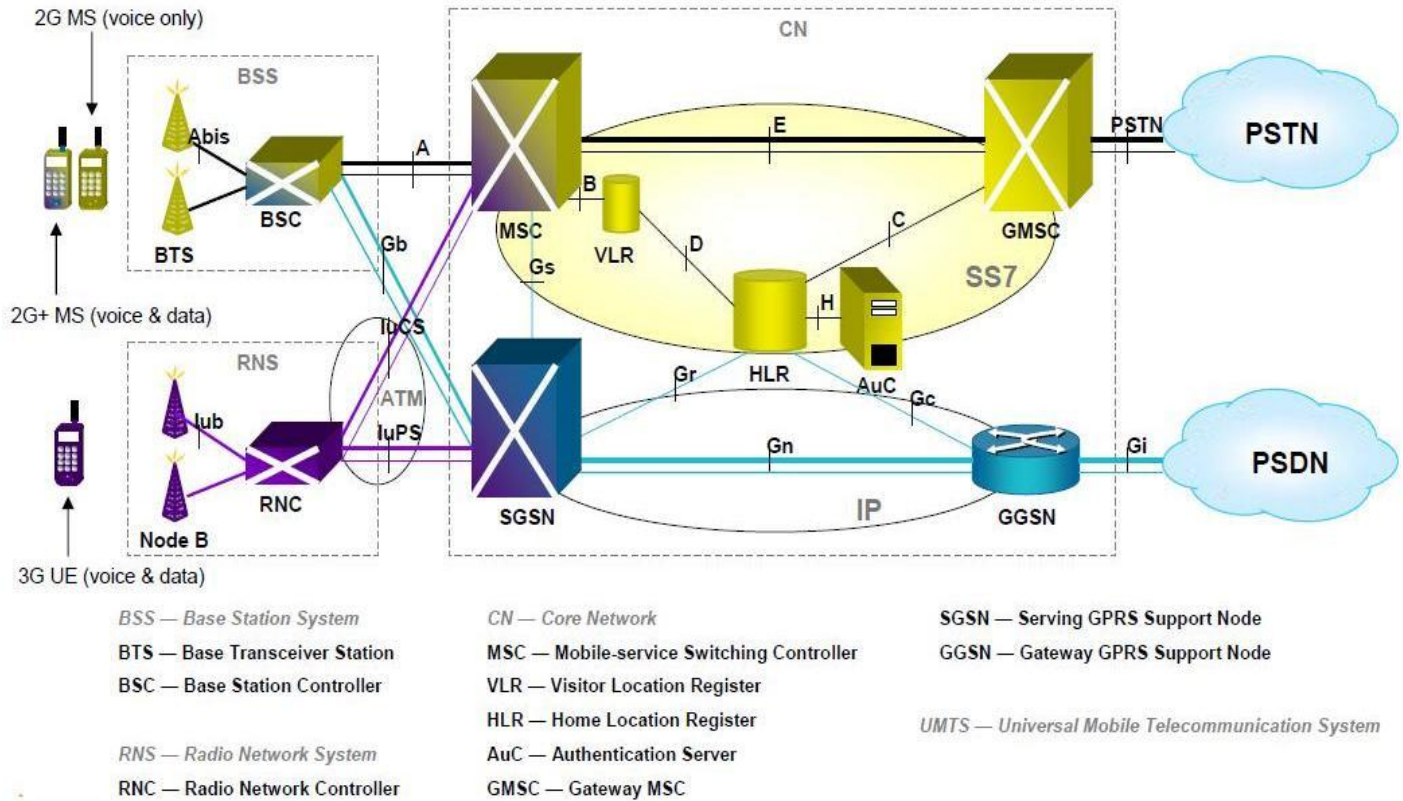
---

### 2.1) OVERVIEW OF 3G ARCHITECTURE

Universal Mobile Telecommunications System (UMTS), standardized by the 3GPP, is the 3G mobile communication technology successor to GSM and GPRS. UMTS combines the W-CDMA, TD-CDMA, or TD-SCDMA air interfaces, GSM's Mobile Application Part (MAP) core, and the GSM family of speech codecs. W-CDMA is the most popular cellular mobile telephone variant of UMTS in use. UMTS, using W-CDMA, supports up to 14.0 Mbit/s data transfer rates in theory with High Speed Downlink Packet Access (HSDPA), although the performance in deployed networks could be much lower for both uplink and downlink connections.

A major difference of UMTS compared to GSM is the air interface forming Generic Radio Access Network (GeRAN). It can be connected to various backbone networks like the Internet, ISDN, and GSM or to a UMTS network. GeRAN includes the three lowest layers of OSI model. The network layer (OSI 3) protocols form the Radio Resource Management protocol (RRM). They manage the bearer channels between the mobile terminals and the fixed network including the handovers.

The UMTS standard is an extension of existing networks based on the GSM and GPRS technologies. In UMTS release 1, a new radio access network UMTS terrestrial radio access network (UTRAN) is introduced. UTRAN, the UMTS radio access network (RAN), is connected via the **Iu** to the GSM Phase 2+ core network (CN). The **Iu** is the UTRAN interface between the radio network controller (RNC) and CN; the UTRAN interface between RNC and the packet-switched domain of the CN (**Iu-PS**) is used for PS data and the UTRAN interface between RNC and the circuit-switched domain of the CN (**Iu-CS**) is used for CS data.



**FIGURE 1: 3G REL99 ARCHITECTURE**

UTRAN is subdivided into individual radio network systems (RNSs), where each RNS is controlled by an **RNC**. The **RNC** is connected to a set of Node B elements, each of which can serve one or several cells. Two new network elements, namely **RNC** and **Node B**, are introduced in **UTRAN**. The **RNC** enables autonomous radio resource management (RRM) by UTRAN. It performs the same functions as the GSM **BSC**, providing central control for the **RNS** elements (**RNC** and Node Bs). **Node B** is the physical unit for radio transmission/reception with cells. **Node B** connects with the **UE** via the W-CDMA **Uu** radio interface and with the **RNC** via the **Iub** asynchronous transfer mode (ATM)-based interface.

## 2.2) 3G SECURITY ARCHITECTURE

The primary reason for the advent of 3G was to provide high end services to numerous users across the globe using a universal handset. However this increased the level of interaction between users, service providers and market operators and also increased the vulnerability of the networks to external attacks.

## MOTIVATION

The UMTS security framework focused on addressing the weaknesses in GSM while enhancing the already successful robust and important methods.

Some of the weaknesses in GSM security architecture are:

- False base station attacks
- Transmission in the open of encryption keys and authentication data
- No encryption provision in the microwave links of the core network
- No integrity protection of data
- No provision for upgrade of security features over time.

3G security provides additional security features and services apart from improving on the above deficiencies of GSM. The aim of 3G security architecture is to build a flexible system adaptive to future changes rather than building a fool proof system.

2G security overlooked several kinds of attacks [8] which 3G security architecture has handled successfully.

To launch these attacks an intruder must have the following capabilities:

- Eavesdropping
- Impersonation of a user
- Impersonation of the network
- Man-in-the-middle attack
- Compromising authentication vectors in the network.

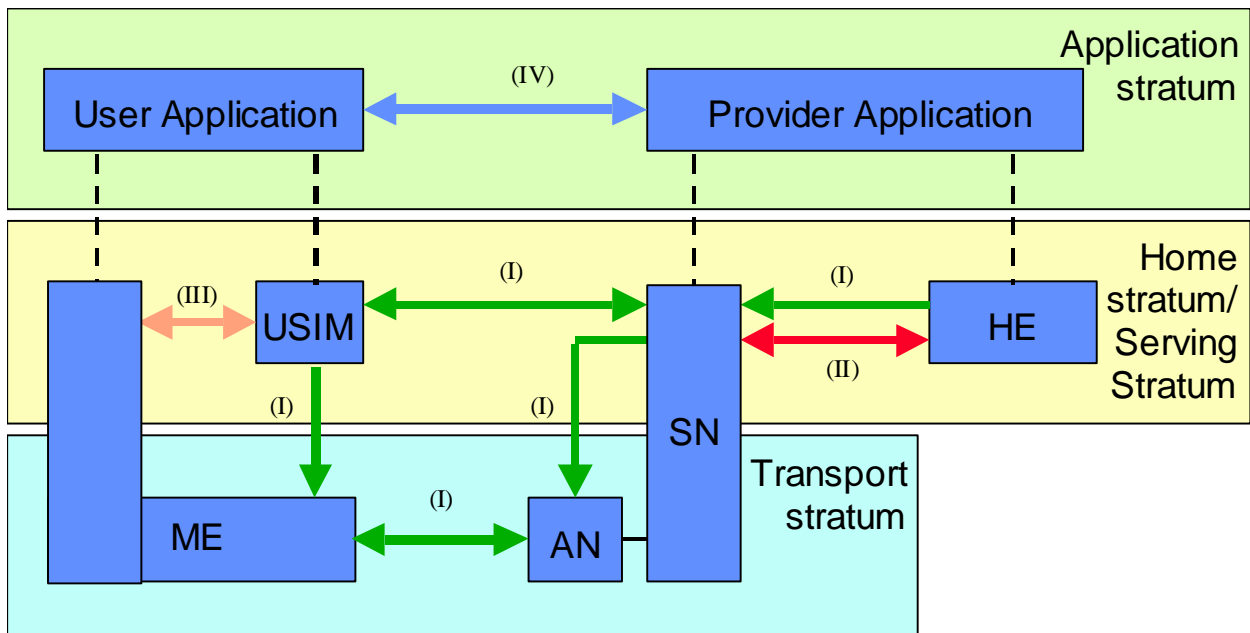
The various types of attacks by a user possessing the above qualities are [8]:

- Denial of service
- Identity catching
- Impersonation of the network and subsequent eavesdropping
- Impersonation of the user

## 2.3) THE UMTS SECURITY ARCHITECTURE

Five distinct security classes are specified by the 3GPP for the 3G security architecture to address certain threats [5] and to provide appropriate security services [7]:

- *Network access security*: ensures confidentiality of user identity and also of the user and signaling data, integrity protection of important signaling data, authentication of user between operator, and identification of Mobile Equipment (ME).
- *Network domain security*: allows various points in the serving network(SN) to exchange signaling data in a secure manner, and prevents attacks on microwave links in the core network.
- *User domain security*: restricts the access to Universal Subscriber Identity Module (USIM) and Mobile Station (MS) to authorized users only [2].
- *Application domain security*: extends security to the application layer ensuring secure communication of applications in the user and service layers.
- *Visibility and configurability of security*: notifies the user of the various security features available and the applicability of these features to various services.



**FIG 2: OVERVIEW OF UMTS SECURITY ARCHITECTURE [7]**

---

## *CHAPTER 3: NETWORK ACCESS SECURITY*

---

This security class provides security features that enable users to securely access 3G services and guards against attacks on the radio interface [7]. Network access security works independently in each service domain. Our work was to implement the network access security. In this work we implemented the MILENAGE algorithm [14, 15] and KASUMI [13] algorithm presented in this chapter.

### 3.1) USER IDENTITY CONFIDENTIALITY

This procedure enables user identification on the radio access link through a Temporary Mobile Subscriber Identity (TMSI)[1]. A TMSI has a local scope only in the area where the user is registered. The Visited Location Register/Service GPRS Support Node (VLR/SGSN) stores the link between the temporary and permanent user identities. To prevent tracing or tracking of user identities, the temporary ids (TMSI) of the user are changed frequently. Further, any signaling or user data that might contain the user's identity are sent in encrypted form on the radio access link.

### 3.2) AUTHENTICATION AND KEY AGREEMENT

The two way authentication between the mobile user and the SN is done using this mechanism with the help of a secret key K. The challenge response protocol is used in this algorithm, and was selected so as to maintain compatibility with the GSM/GPRS security architecture helping the transgression from GSM/GPRS to UMTS. In addition, the User Service Identity Module (USIM) and the HE maintain counters SQNMS and SQNHE respectively, which are used in network authentication. Each user maintains its own counter SQNHE, while the counter SQNMS stores the highest sequence number accepted by the USIM [1].

The VLR/SGSN requests the HE Authentication Center (HE/AuC) to send the next ordered array of Authentication Vectors (AV) to it. Each AV contains an unpredictable challenge viz. a expected response XRES, a confidentiality key CK, an integrity key IK, an random number RAND and an authentication token AUTN and is implemented in the authentication and key agreement mechanism between the VLR/SGSN and the USIM.

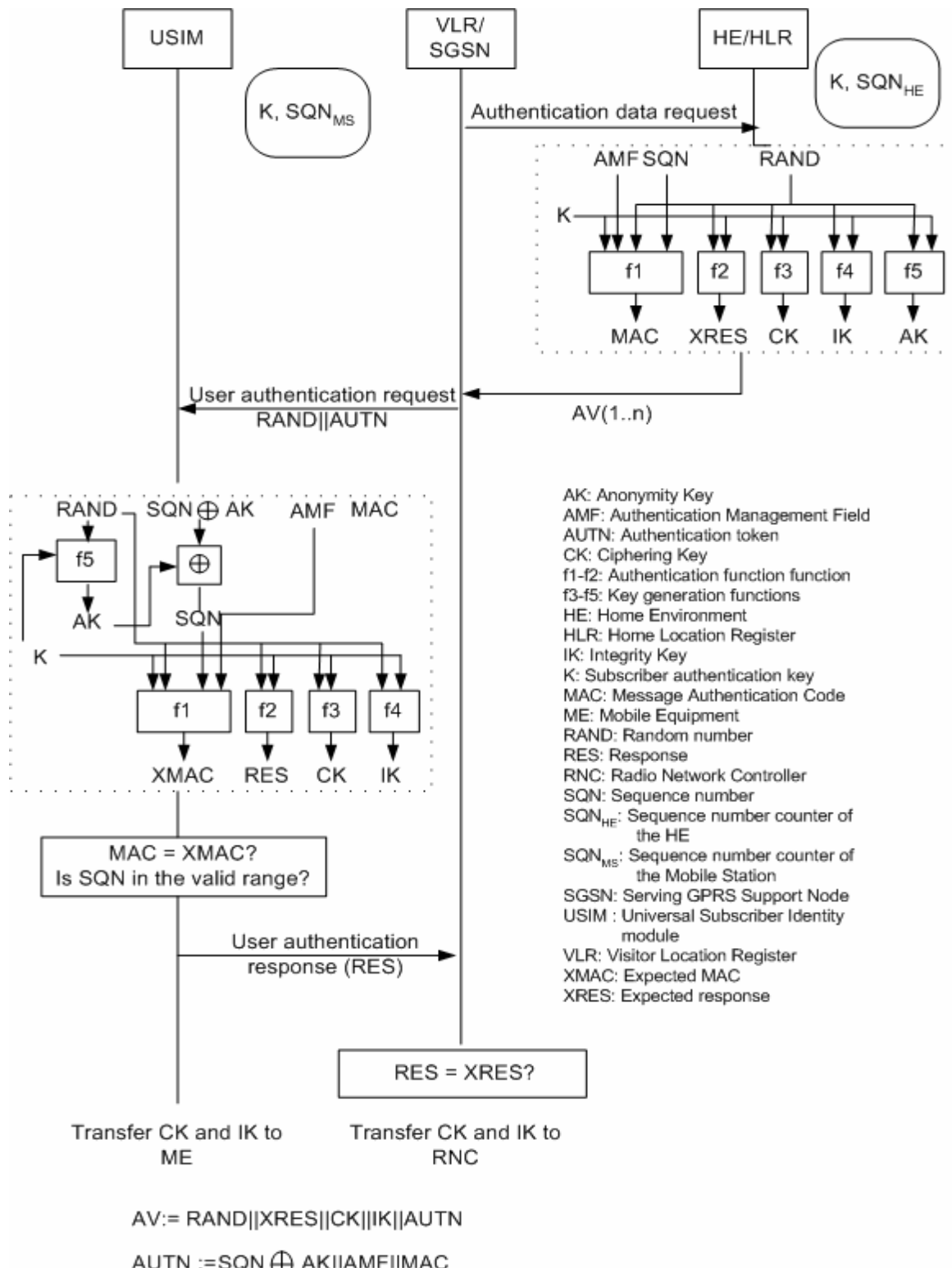
The HE/AuC first generates a unused sequence number SQN and an unpredictable challenge RAND [1]. Then it computes the following functions:

- The Message Authentication Code  $MAC = f1k (SQN // RAND // AMF)$ , where  $f1$  is a message authentication function, and the Authentication and key Management Field (AMF) is applied for performance optimization, or to select a new authentication key from the USIM [7,4].
- The expected response  $XRES = f2k (RAND)$  where  $f2$  is the message authentication function.
- The Cipher Key  $CK = f3k (RAND)$ ,
- The Integrity Key  $IK = f4K (RAND)$ ,
- The Anonymity Key  $AK = f5K (RAND)$  where  $f3$ ,  $f4$  and  $f5$  are key generating functions.
- Finally, the HE/AuC combines the authentication token  

$$AUTN = SQN \oplus AK // AMF // MAC.$$

The VLR/SGSN starts the authentication and key agreement mechanism by selecting a new AV from the ordered array, and sends the parameters RAND and AUTN to the user. The USIM computes the AK,  $AK = f5K (RAND)$ , and then extracts the SQN by  $SQN = (SQN \oplus AK) \oplus AK$ . Then, it generates  $XMAC = f1K (SQN // RAND // AMF)$ , and verifies that the received AUTN and the retrieved SQN values are within satisfactory range [7] (see fig 3).

If the above condition satisfies then USIM computes the  $RES = f2K (RAND)$ , and sends back a user authentication response through the MS. Then the USIM calculates the CK,  $CK = f3K (RAND)$  and the IK,  $IK = f4K (RAND)$ . The VLR/SGSN checks the received RES with the XRES field of the AV. If they are same, then the authentication and key agreement exchange is declared a success. In the end , the USIM and the VLR/SGSN send the generated keys, CK and IK, to the mobile equipment and the Radio Network Controller (RNC) that perform ciphering and integrity functions.



**FIG 3: AUTHENTICATION AND KEY AGREEMENT PROCEDURE [7]**



## 3.3) MILENAGE ALGORITHM

### 3.3.1) INTRODUCTION

The MILENAGE algorithm set [14,15] was developed by the 3GPP Task Force and meant to be used as an example set for authentication and key agreement procedure [7]. It consists of seven functions  $f_1$ ,  $f_1^*$ ,  $f_2$ ,  $f_3$ ,  $f_4$ ,  $f_5$  and  $f_5^*$  which may be used as authentication and key generating functions. This algorithm is not standardized rather it is provided as an example set for operators to use if they do not want design an algorithm of their own. All seven functions are operator specific.

The functions used in authentication and key agreement [14] are:

- $f_0$ : the random challenge generating function
- $f_1$ : the network authentication function
- $f_1^*$ : the resynchronization message authentication function
- $f_2$ : the user authentication function
- $f_3$ : the cipher key derivation function
- $f_4$ : the integrity key derivation function
- $f_5$ : the anonymity key derivation function
- $f_5^*$ : the anonymity key derivation function for resynchronization

### 3.3.2) KEY FEATURES

- **Resilience:** The functions are designed so that they can withstand continuous attacks for a duration of not less than 20 years.
- The algorithm fulfils all the requirements specified in 3G TS 33.105 [14].
- The algorithm can be personalized based on an 128 bit operator variant configuration algorithm field.
- The kernel function used in the algorithm uses standard/publicly available algorithms.
- It can successfully counterattack Differential Power Analysis, Simple Power Analysis, and other 'side-channel' attacks when implemented on a USIM.

- The functions  $f1$ ,  $f1^*$ ,  $f2$ ,  $f3$ ,  $f4$ ,  $f5$  and  $f5^*$  are nearly identical from independent random functions of their inputs (RAND||SQN||AMF) and RAND without getting a hold on secret keys.
- It is nearly impossible to derive any portion of the secret key  $K$ , or the operator specific parameter  $OP$ , by examining the inputs and the outputs to the algorithm.
- Events tending to violate above criteria occur with probability approximately  $2^{-128}$

### 3.3.3) PARAMETERS USED [15]

AK	A 48 bit anonymity key generated by the functions $f5$ and $f5^*$
AMF	A 16-bit authentication management field given as input to the functions $f1$ and $f1^*$ .
c1, c2, c3, c4, c5	Arbitrary constants which are 128-bits in length and XORed into intermediate variables.
CK	A 128-bit confidentiality key generated by the function $f3$ as output.
IK	A 128-bit integrity key generated by the function $f4$ as output.
INI	A 128-bit value derived from SQN and AMF and implemented in the calculation of the functions $f1$ and $f1^*$ .
K	A 128-bit subscriber key that acts as input to the functions $f1$ , $f2$ , $f5^*$ , $f3$ , $f1^*$ , $f5$ and $f4$ .
MAC-A	A 64-bit network authentication code generated by the function $f1$ as output.
MAC-S	A 64-bit resynchronization authentication code generated by the function $f1^*$ as output.
OP	A 128-bit Operator Variant Algorithm Configuration Field that is a part of the functions $f1$ , $f2$ , $f5^*$ , $f3$ , $f1^*$ , $f5$ and $f4$ .
OPc	A 128-bit value derived from $OP$ and $K$ and used inside the implementation of the functions.
OUT1,OUT2,OUT3,OUT4,OUT5	128-bit calculated values that are used to generate the outputs of the functions $f1$ , $f2$ , $f5^*$ , $f3$ , $f1^*$ , $f5$ and $f4$ .
r1, r2, r3, r4, r5	Integers varying from 0 to 127 both included, which specify the degree of cyclic rotation of intermediate variables.
RAND	A 128-bit unpredictable random challenge given as input to the functions $f1$ , $f2$ , $f5^*$ , $f3$ , $f1^*$ , $f5$ and $f4$ .
RES	A 64-bit parameter generated by the function $f2$ as output and used as response.
SQN	A 48-bit sequence number that given as input to either $f1^*$ or $f1$ . It is better known as $SQN_{MS}$ in $f1^*$ .
TEMP	A 128-bit value used within the computation of the functions to store temporary values.

The algorithm makes use of the following two components:

- A block cipher encryption function, which inputs a 128-bit variable and generates a 128-bit output using a key of length 128 bits.
- A 128-bit value **OP**. **OP** or Operator Variant Algorithm Configuration Field provides uniqueness to the algorithms when used by different operators. This parameter is operator specific i.e Each operator can choose its own **OP**. The algorithm set is secure whether or not **OP** is known

### 3.3.4) ALGORITHM FRAMEWORK

$OPC(128 \text{ bits})$  is obtained from **OP** and **K** as shown [15]:

$OPC = OP \oplus E [OP] K$ . The intermediate value **TEMP** (128 bits) is calculated as follows:

$$TEMP = E [RAND \oplus OPC] K. \quad e$$

**IN1** which is 128 bits in length is derived as follows:

$$IN1 [0] \dots IN1 [47] = SQN [0] \dots SQN [47]$$

$$IN1 [48] \dots IN1 [63] = AMF [0] \dots AMF [15]$$

$$IN1 [64] \dots IN1 [111] = SQN [0] \dots SQN [47]$$

$$IN1 [112] \dots IN1 [127] = AMF [0] \dots AMF [15] \quad c1,$$

**c2, c3, c4, c5** are arbitrary constants of 128 bits as defined here:

$$c1 [i] = 0 \text{ for } 0 \leq i \leq 127$$

$$c2 [i] = 0 \text{ for } 0 \leq i \leq 127, \text{ except that } c2 [127] = 1$$

$$c3 [i] = 0 \text{ for } 0 \leq i \leq 127, \text{ except that } c3 [126] = 1$$

$$c4 [i] = 0 \text{ for } 0 \leq i \leq 127, \text{ except that } c4 [125] = 1$$

$$c5 [i] = 0 \text{ for } 0 \leq i \leq 127, \text{ except that } c5 [124] = 1$$

**r1, r2, r3, r4, r5** are integers which are arbitrary. They are defined here:

$$r1 = 64; r2 = 0; r3 = 32; r4 = 64; r5 = 96$$

Five 128-bit blocks **OUT1, OUT2, OUT3, OUT4 and OUT5** are computed as follows:

$$OUT1 = E [TEMP \oplus \text{rot} (IN1 \oplus OP_C, r1) \oplus c1]_K \oplus OP_C$$

$$OUT2 = E [\text{rot} (TEMP \oplus OP_C, r2) \oplus c2]_K \oplus OP_C$$

$$OUT3 = E [\text{rot} (TEMP \oplus OP_C, r3) \oplus c3]_K \oplus OP_C$$

$$OUT4 = E [\text{rot} (TEMP \oplus OP_C, r4) \oplus c4]_K \oplus OP_C$$

$$\mathbf{OUT5} = E [\text{rot} (\mathbf{TEMP} \oplus \mathbf{OP}_C, \mathbf{r5}) \oplus \mathbf{c5}]_K \oplus \mathbf{OP}_C$$

The outputs of the various functions are derived here:

Output of  $f1$  = MAC-A, where MAC-A[0] .. MAC-A[63] =  $\mathbf{OUT1}[0] \dots \mathbf{OUT1}[63]$

Output of  $f1^*$  = MAC-S, where MAC-S [0] .. MAC-S[63] =  $\mathbf{OUT1}[64] \dots \mathbf{OUT1}[127]$

Output of  $f2$  = RES, where RES [0] .. RES [63] =  $\mathbf{OUT2} [64] \dots \mathbf{OUT2} [127]$

Output of  $f3$  = CK, where CK [0] .. CK [127] =  $\mathbf{OUT3} [0] \dots \mathbf{OUT3} [127]$

Output of  $f4$  = IK, where IK [0] .. IK [127] =  $\mathbf{OUT4} [0] \dots \mathbf{OUT4} [127]$

Output of  $f5$  = AK, where AK [0] .. AK [47] =  $\mathbf{OUT2} [0] \dots \mathbf{OUT2} [47]$

Output of  $f5^*$  = AK, where AK [0] .. AK [47] =  $\mathbf{OUT5} [0] \dots \mathbf{OUT5} [47]$

### 3.3.5) IMPLEMENTATION CONCERNS

There are two implementations considerations this algorithm:

- $\mathbf{OP}_C$  computed on or off the USIM [15]
- Choice of Block Cipher.

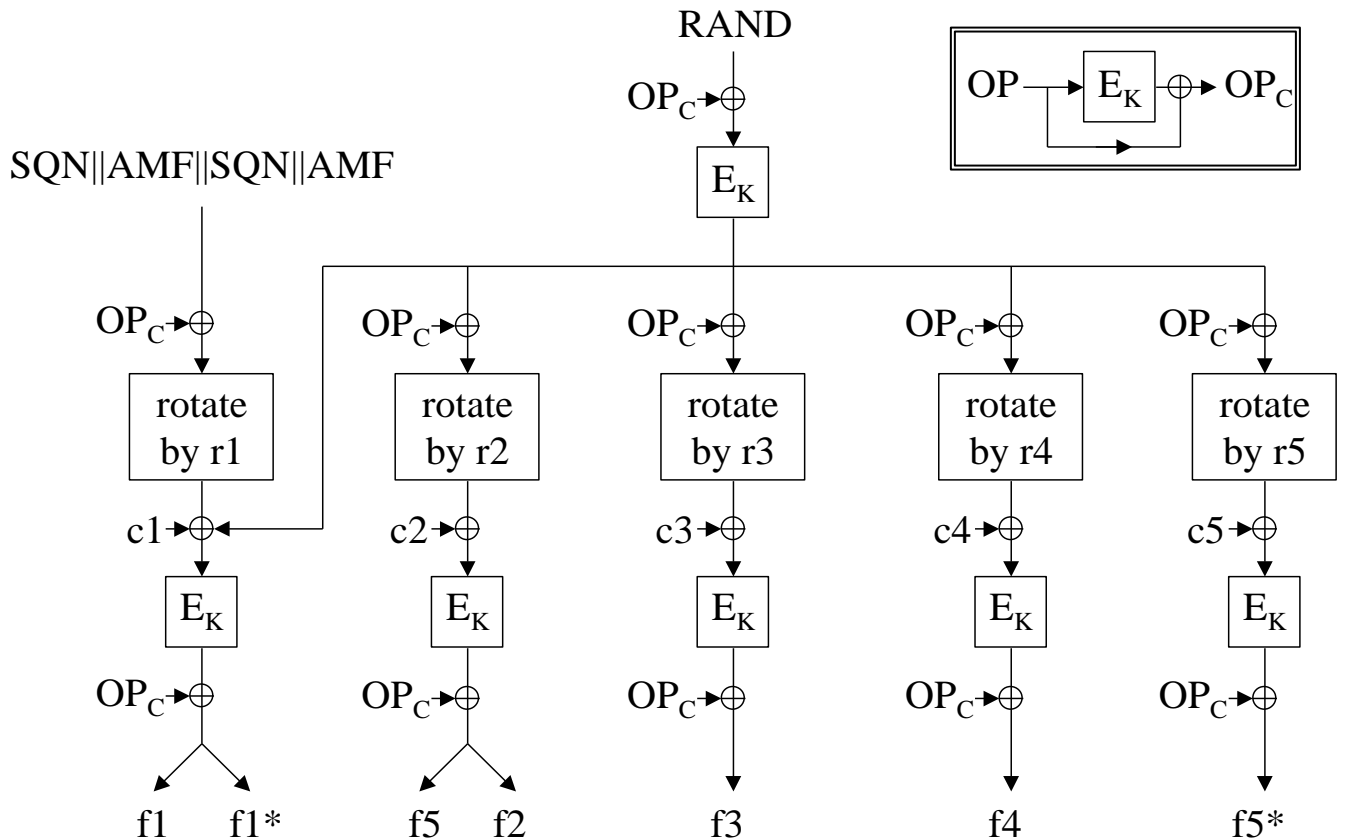
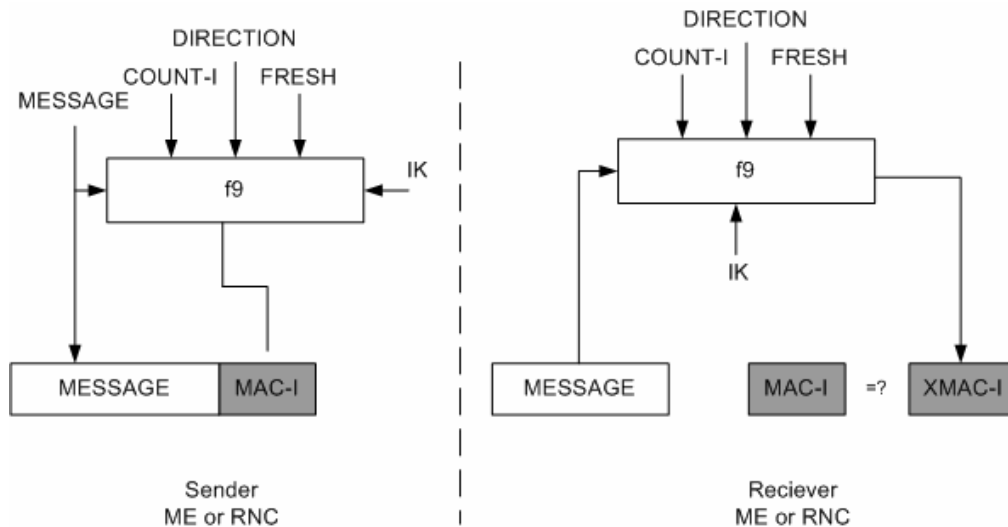


FIG 4: DEFINITION OF F1, F1\*, F2, F3, F4, F5 AND F5\*[15]

### 3.4) INTEGRITY PROTECTION OF SIGNALING MESSAGES

The radio access interface in 3G mobile systems are designed such that they support integrity protection on the signaling channels so that the receiving nodes (MS or SN) can ensure that the signaling data have not been changed or tampered with illegally on the way from the sender [1]. It also ensures that the source of the received signaling data is authentic. The integrity protection guards against false base station attacks, and prevents potential intruders from hijacking connections in the absence of any ciphering [8]. The function  $f_9$  is used to ensure the integrity and the source of signaling data between the RNC and the ME in 3G security framework. It generates a 32-bit Message Authentication Code (MAC) that is attached to the end of the frame, and is matched by the receiver (see fig 5),.

The primary inputs to the algorithm are a 128-bit secret Integrity key  $IK$ , and the frame content  $MESSAGE$  which can have any length. Additional inputs, which ensure that two frames with identical data have unique MACs, are a 32-bit value  $FRESH$ , a 32-bit value  $COUNT$  and a 1-bit value  $DIRECTION$ . The UMTS release '99 architecture has the  $f_9$  based on the Kasumi algorithm [13].

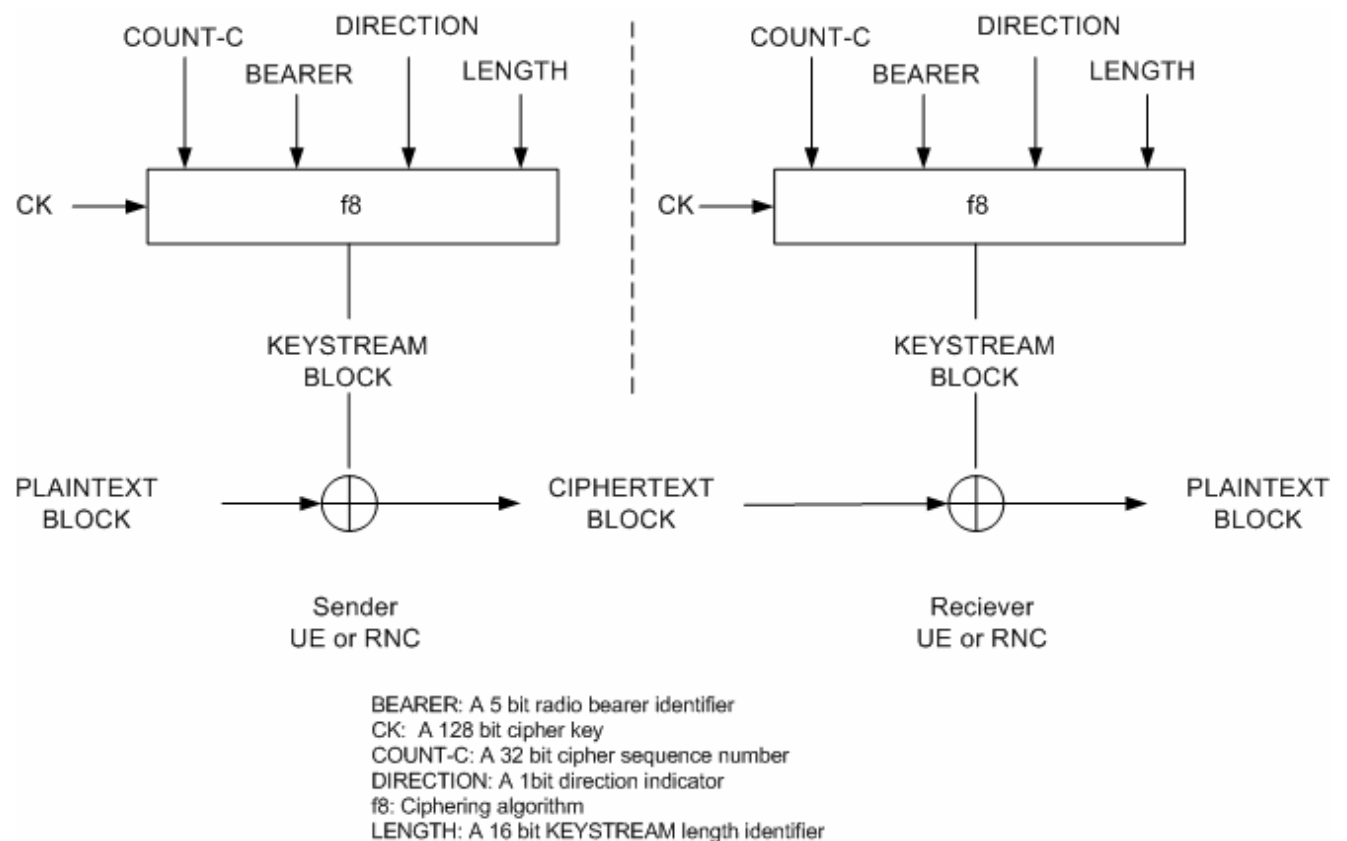


COUNT-I: A 32 bit integrity sequence number  
 DIRECTION: A 1 bit direction indicator  
 FRESH: A network side nonce  
 IK: A 128bit integrity key  
 f9: Integrity algorithm  
 MESSAGE: The message to be protected  
 XMAC-I: Expected MAC-I

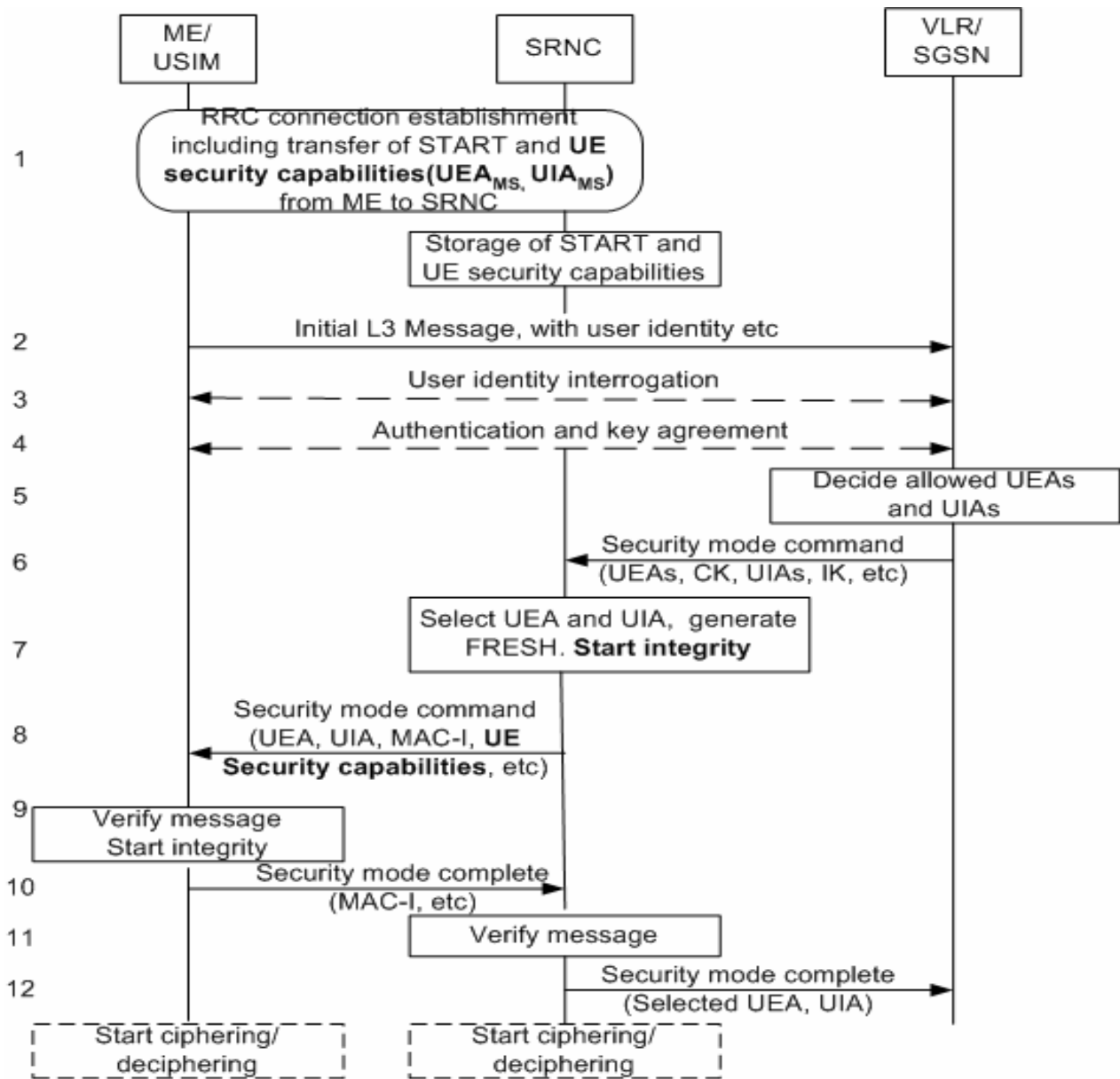
**FIG 5: DERIVATION OF MAC (OR XMAC) ON A SIGNALING MESSAGE [12]**

### 3.5) DATA CONFIDENTIALITY

User and signaling data sent over the radio interface, are subjected to encryption using the function f8 [1]. The f8 is a symmetric synchronous stream cipher used for ciphering frames of different length. The primary input to the f8 is a 128-bit secret Cipher Key CK. To apply uniqueness to frames such that they are encrypted using two different keystreams, a 5-bit value BEARER, a 32-bit value COUNT and a 1-bit value DIRECTION are applied. The output is a stream of bits (keystream) having length equal to that of the frame. Then the data is XORed with the keystream for encryption of the frame. The UMTS release '99 has the f8 function based on the Kasumi algorithm [12, 13].



**FIG 6: CIPHERING OVER RADIO ACCESS LINK [12]**



FRESH: A network side nonce  
 ME: Mobile equipment  
 UE: User equipment

START  
 MAC-I:  
 SRNC: Serving Radio Network Controller

UIA: UMTS Integrity algorithm  
 UEA: UMTS Encryption algorithm

**FIG 7: OVERALL SETUP OF 3G SECURITY [2, 7]**

## 3.6) KASUMI ALGORITHM

### 3.6.1) INTRODUCTION

The 3GPP security architecture specifies two standardized algorithms: A confidentiality algorithm  $f8$ , and an integrity algorithm  $f9$  [12] both of which use the **KASUMI** algorithm [13]. **KASUMI** is a block cipher that takes a 64-bit input and a 128-bit key and generates a 64-bit output.

### 3.6.2) LIST OF VARIABLE [12]

A, B	64-bit registers used within the $f8$ and $f9$ functions to store intermediate values.
BEARER	A 5-bit input to the $f8$ function
BLOCKS	An integer variable specifying the number of successive operations of <b>KASUMI</b> for both the $f8$ and $f9$ functions.
BLKCNT	A 64-bit counter used in the $f8$ function
FRESH	A 32-bit random input to the $f9$ function
DIRECTION	A 1-bit input to both the $f8$ and $f9$ functions denoting the direction of transmission (uplink or downlink).
IBS	The bit stream used as input to the $f8$ function
KM	A 128-bit constant which acts as a key modifier in both the $f8$ and $f9$ functions. However the value in each function is different.
IK	A 128-bit integrity key.
KS[i]	The $i^{\text{th}}$ bit of key stream generated by the key stream generator
KSB <sub><i>i</i></sub>	The $i^{\text{th}}$ block of keystream generated by the keystream generator. All the blocks of keystream are of 64 bits.
LENGTH	An input to the $f8$ and $f9$ functions. It contains the number of bits in the input bitstream
MAC-I	The 32-bit message authentication code (MAC) generated by the integrity function $f9$ .
MESSAGE	The input bitstream of LENGTH bits that is to be computed by the $f9$ function
OBS	The output bit streams obtained using the $f8$ function
PS	The input padded string used in the $f9$ function.
REGISTER	A 64-bit value that implemented inside the $f8$ function



### 3.6.3) CONFIDENTIALITY ALGORITHM F8

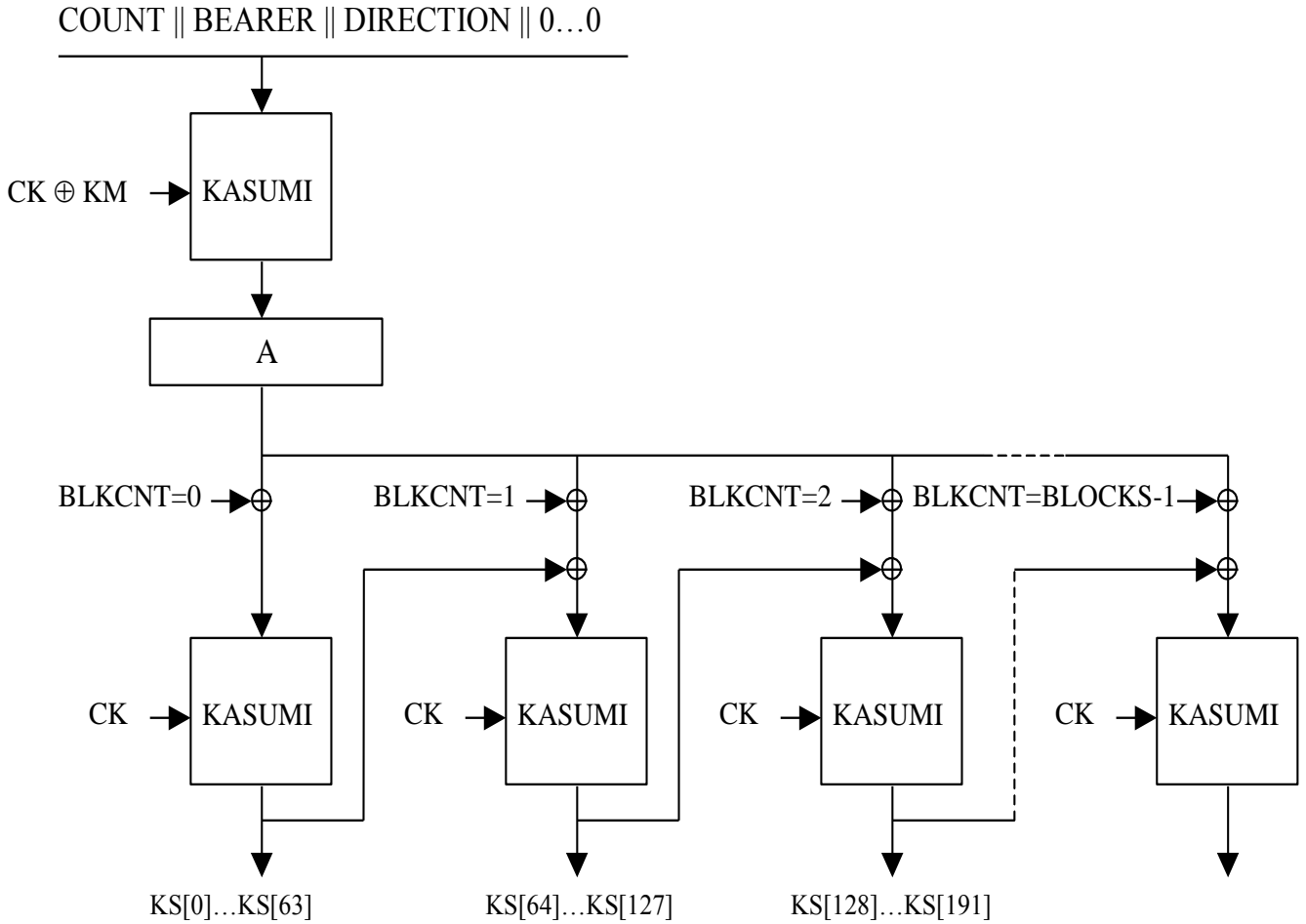
The confidentiality algorithm *f8* is used for encryption/decryption of blocks of data using a confidentiality key **CK** [12]. It is basically a stream cipher. The length of the block of data can be between 1 and 20000 bits. **KASUMI** is used in this algorithm as a keystream generator in output-feedback mode and gives the output keystream in blocks of 64-bits. The feedback data is changed by static data stored in a 64-bit register **A**, and an (increasing) 64-bit counter **BLKCNT**.

**TABLE 1: F8 INPUT [12]**

Parameter	Size(bits)	Comment
COUNT	32	Frame dependent input COUNT[0].....COUNT[31]
BEARER	5	Bearer identify BEARER[0].....BEARER[4]
DIRECTION	1	Direction of transmission DIRECTION[0]
CK	128	Confidentiality key CK[0]....CK[127]
LENGTH		The number of bits to be Encrypted/Decrypted
IBS	1-20000	Input bit stream IBS[0]....IBS[LENGTH-1]

**TABLE 2: F8 OUTPUT [12]**

Parameter	Size(bits)	Comment
OBS	1-20000	Output bit stream OBS[0].....OBS[LENGTH-1]



**FIG 8: F8 KEYSTREAM GENERATOR [12]**

**INITIALIZATION**

The 64-bit register **A** is set to **COUNT || BEARER || DIRECTION || 0...0** (left justified with the right most 26 bits set to 0).i.e.

$$A = \text{COUNT}[0] \dots \text{COUNT}[31] \text{ BEARER}[0] \dots \text{BEARER}[4] \text{ DIRECTION}[0] 0 \dots 0.$$

The counter **BLKCNT** is set to zero.

The key modifier **KM** is set to 0x55555555555555555555555555555555,

Then the **KSB<sub>0</sub>** is set to zero [12].

A refined version of the confidentiality key as shown here is used in one instance of **KASUMI** and employed to the register **A**.

$$A = \text{KASUMI} [ A ]_{\text{CK} \oplus \text{KM}}$$

## KEYSTREAM GENERATION

After the keystream generator has been initialized as defined above, it can be used to produce keystream bits [12]. The plaintext/ciphertext used in encryption/decryption contains **LENGTH** bits between 1 and 20000 while the keystream generator generates keystream bits in multiples of 64 bits. The least significant bits (0-63) are rejected from the last block basing on the total number of bits needed by **LENGTH**. In our case, **BLOCKS** is set equal to  $(\text{LENGTH}/64)$  rounding up to the nearest integer. (For example, if **LENGTH** = 128 then **BLOCKS** = 2; if **LENGTH** = 129 then **BLOCKS** = 3).

To output each keystream block (**KSB**) the following operation is performed:

For each integer **n** with  $1 \leq n \leq \text{BLOCKS}$  :

$$\text{KSB}_n = \text{KASUMI}[A \oplus \text{BLKCNT} \oplus \text{KSB}_{n-1}]_{\text{CK}}$$

where  $\text{BLKCNT} = n-1$

The individual bits of the keystream are derived from **KSB**<sub>1</sub> to **KSB**<sub>BLOCKS</sub> in turn, most significant bit first, by using the following operation:

For **n** = 1 to **BLOCKS** and for each integer **i** with  $0 \leq i \leq 63$  :  $\text{KS} [((n-1)*64) + i] = \text{KSB}_n[i]$ .

## ENCRYPTION/DECRYPTION

Encryption/decryption operations are similar and are done by the exclusive-OR of the input data (IBS) with the generated keystream (KS) [12].

For each integer **i** with  $0 \leq i \leq \text{LENGTH}-1$  we define:

$$\text{OBS}[i] = \text{IBS}[i] \oplus \text{KS}[i]$$

### 3.6.4) INTEGRITY ALGORITHM F9

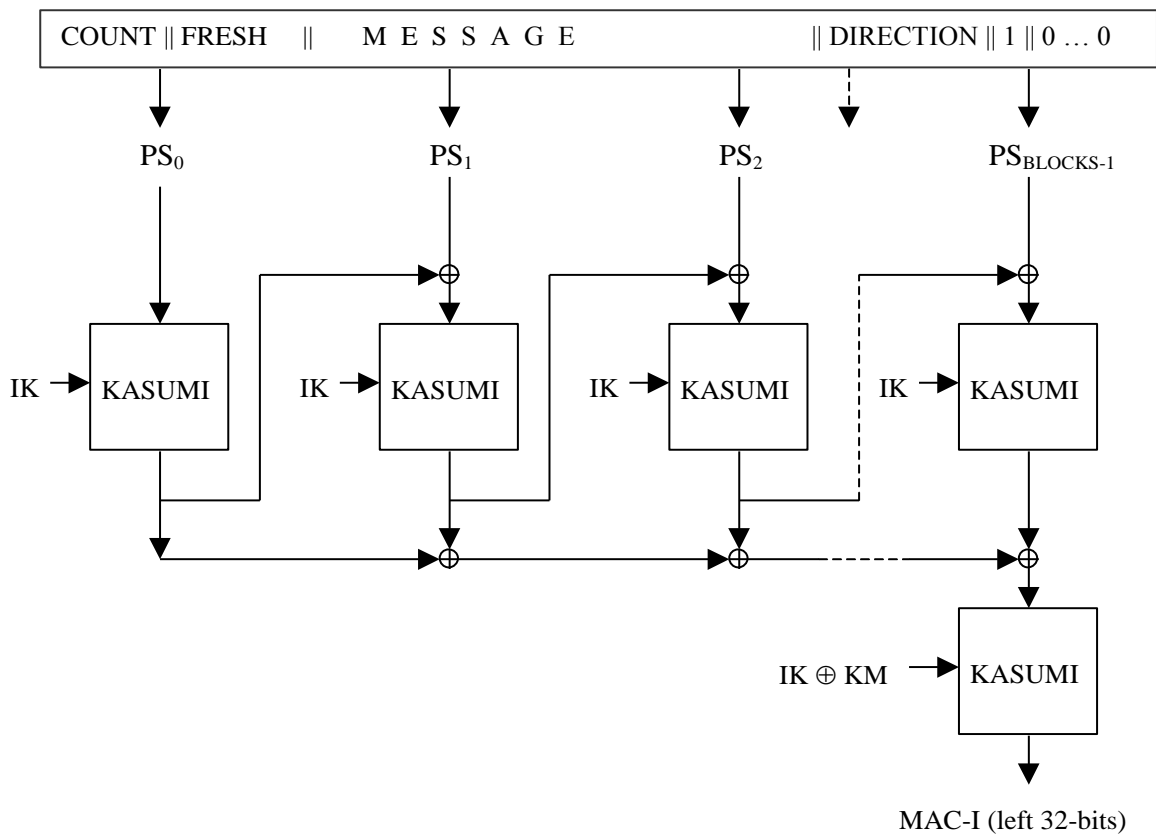
The integrity algorithm **f9** generates a Message Authentication Code (MAC) on an input message using an integrity key **IK**. There is no limit of size on the input message length of the **f9** algorithm. The algorithm uses **KASUMI** [13] block cipher in a form of CBC-MAC mode.

**Table 1:f9 Input [12]**

Parameter	Size(bits)	Comment
COUNT-I	32	Frame dependent input COUNT[0]....COUNT[31]
FRESH	32	Random number FRESH[0].....FRESH[31]
DIRECTION	1	Direction of transmission DIRECTION[0]
IK	128	Integrity key IK[0]...IK[127]
LENGTH	X-19	The number of bits to be 'MAC' d
MESSAGE	LENGTH	Input bit streams

**Table 2:f9 Output [12]**

Parameter	Size(bits)	Comment
MAC-I	32	Message authentication code MAC-I[0].....MAC-I[31]



**FIG 9: F9 INTEGRITY FUNCTION [12]**

**KASUMI** is used in a chained mode to produce a 64-bit intermediate of the message input. At the last, the leftmost 32-bits of the intermediate are taken as the output value **MAC-I**.

#### INITIALIZATION

The integrity function is initialized with the key variables before the calculation begins. The running variables **A** and **B** are set to zero and the key modifier **KM** is set to **KM=0xAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA**.

The variables **COUNT**, **FRESH**, **MESSAGE** and **DIRECTION** are then concatenated [12]. Then a single '1' bit is appended, succeeded by between 0 and 63 '0' bits such that the total length of the generated string **PS** (padded string) is in integral multiples of 64 bits, viz: **PS=COUNT[0]...COUNT[31]FRESH[0]...FRESH[31]MESSAGE[0]...MESSAGE[LENGTH-1]DIRECTION[0]10\***. Here 0\* indicates between 0 and 63 '0' bits.

#### CALCULATION

The padded string **PS** is split into 64-bit blocks **PS<sub>i</sub>** where:

$$PS = PS_0 \parallel PS_1 \parallel PS_2 \parallel \dots \parallel PS_{BLOCKS-1}$$

The following steps are then applied for each integer **n** with  $0 \leq n \leq \mathbf{BLOCKS-1}$ :

$$A = \mathbf{KASUMI} [A \oplus PS_n] \mathbf{IK}$$

$$B = B \oplus A$$

Finally one more operation of **KASUMI** is done using a changed version of the integrity key **IK**.

$$B = \mathbf{KASUMI} [B]_{\mathbf{IK} \oplus \mathbf{KM}}$$

The 32-bit **MAC-I** consists of 32 bits which are left-most in the result.

**MAC-I** = lefthalf [ **B** ] i.e. For each integer **i** with  $0 \leq i \leq \mathbf{31}$  **MAC\_I** is defined as:

$$\mathbf{MAC-I}[i] = B[i] .$$

Bits **B[32]...B[63]** are rejected [12].

---

## *CHAPTER 4: IMPLEMENTATION DETAILS*

---

### 4.1) AUTHENTICATION AND KEY AGREEMENT (AKA)

We implemented the example set of MILENAGE algorithm[15] to establish the authentication and key agreement[7, 14] between the USIM and VLR/SGSN .The authentication and key agreement mechanism in 3G security framework has been described in section 3.2 of this thesis.

The programming language used is C.

To simulate the real life situation on two machines we used socket programming to represent the USIM and AuC as client and server respectively. All communication was done between client and server programs residing on two different machines.

The block cipher used in the kernel function is Rijndael [15].The Rijndael block cipher is based on AES. Rijndael is an block cipher using iteration and having key length and block length of variable size. The block length and the key length can be of 128, 192 or 256 bits in length. In our case, Rijndael has the block length and key length equal to 128 bits and is used only for encryption.

The client and server shared a symmetric key through secret procedure.

The AuC initiates the procedure by selecting an array of authentication vectors. Each AV consists of a unpredictable challenge RAND, and expected XRES, cipher key CK and integrity key IK and authentication token AUTN. The AuC forwards the parameters RAND and  $AUTN(SQN \oplus AK || AMF || MAC)$  to the user. The USIM computes the AK using the secret key K. Then it calculates the  $XMAC=f_{1k}(SQN || RAND || AMF)$  and verifies whether the received AUTN and the retrieved SQN values originated in the AuC [1,7].

If the above condition satisfies then the USIM calculates the  $RES = f_{2k}(RAND)$  and asks the mobile station to send back a user authentication response. After the USIM computes the CK and the IK, the VLR/SGSN checks the received RES with the XRES field of the AV. If they are the same then the authentication and key agreement procedure is declared as successfully completed.

## 4.2) SOCKET PROGRAMMING

We now give a brief introduction to socket programming in c and specify the functions used for our purpose.

A socket is an Application Programming Interface (API) used for Inter Process Communication (IPC).[A well defined method of connecting two processes locally or across a network].It is protocol and language independent and is often referred to as Berkeley Sockets or BSD Sockets.

TWO IMPORTANT PROTOCOLS:

TCP/IP-Provides reliable in-order transfer of bytes between client and server.

UDP-Provides unreliable transfer of groups of bytes between server and client.

### PRIMARY SOCKET CALLS

socket()-creates a new socket and returns it descriptor.

bind()-associates a socket with a port and address.

listen()-establish a queue for connection request.

accept()-accepts a connection request.

connect()-initiate a connection to a remote host.

recv()-receives data from socket descriptor.

send()-sends data to a socket descriptor.

close()-“one way” close of a socket descriptor.

### PRIMARY HEADER FILES

Include file sequence may affect processing(order is important!)

<sys/types.h>-prerequisite typedefs

<errno.h>names for “errno” values (error numbers)

<sys/socket.h>-struct sockaddr;system prototypes and structures.

<netdb.h>-network info lookup prototypes and structures  
<netinet/in.h>-struct sockaddr\_in; byte ordering macros  
<arpa/inet.h>-utility function prototypes.

### 4.3) CONFIDENTIALITY AND INTEGRITY

We implemented the confidentiality algorithm **f8** for data confidentiality and the integrity algorithm **f9** using the example algorithm set in Annex 2[12, 13]. All these algorithms use the **KASUMI** algorithm [13].

The programming language used is C.

The block cipher used is Kasumi. **KASUMI** is a block cipher that takes a 64-bit input and generates a 64-bit output using a 128-bit key.

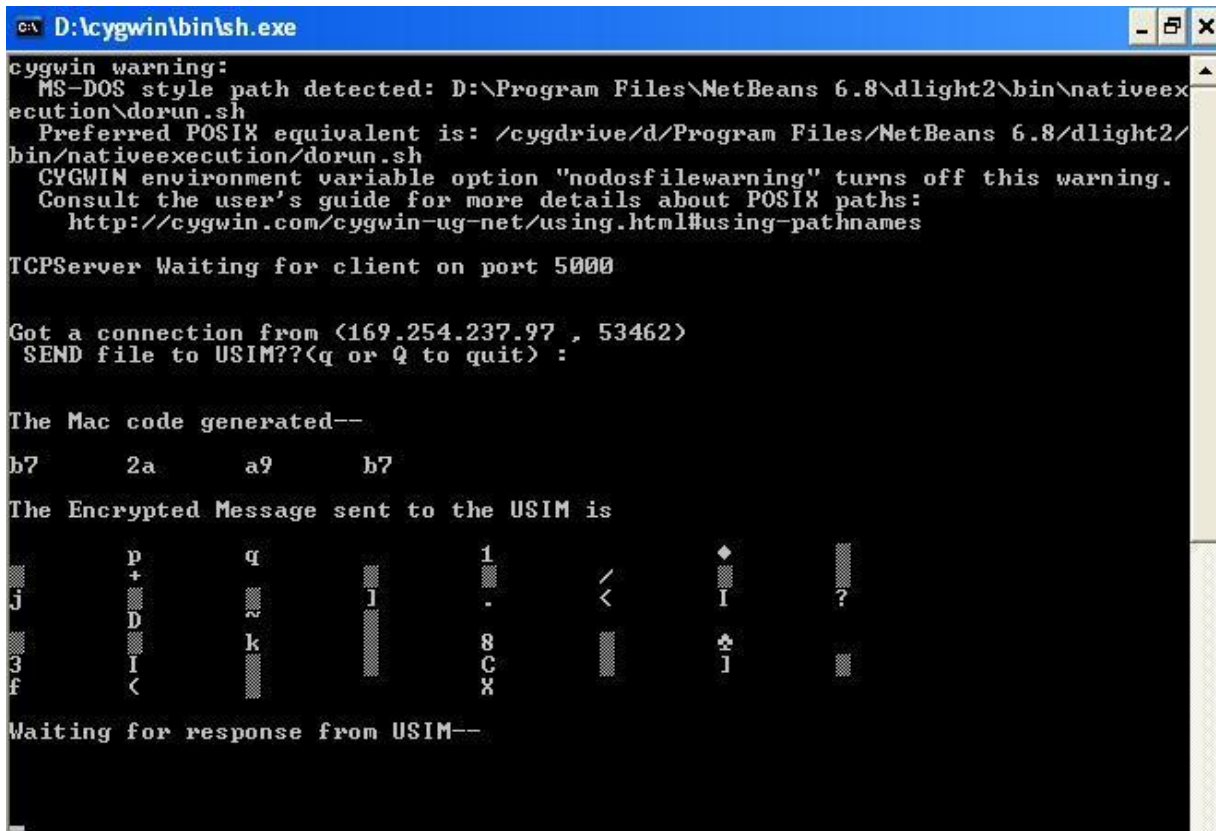
We used socket programming as before to simulate MS and RNC as client and server respectively. All communication was done between client and server programs residing on two different machines. From here on we will refer to MS and RNC as client as server respectively.

In addition we used the concepts of file handling to send files in encrypted form from server to client and vice versa.

The server encrypts the source file using the function **f8** and stores it in another file. It then computes the 32 bit MAC code and appends it to the end of the file. The file is sent to the client using socket connection. The client extracts the MAC code. It calculates its own MAC code from previously generated **IK** and checks the calculated MAC with the received MAC. If they match then the integrity of the incoming message is verified. If so, then the receiver decrypts the incoming message using the function **f8** and **CK** generated before. The same process is applied when the client sends a message to the server.



## 4.4) SCREENSHOTS



```
D:\cygwin\bin\sh.exe
cygwin warning:
MS-DOS style path detected: D:\Program Files\NetBeans 6.8\dlight2\bin\nativeex
ecution\dorun.sh
Preferred POSIX equivalent is: /cygdrive/d/Program Files/NetBeans 6.8/dlight2/
bin/nativeexecution/dorun.sh
CYGWIN environment variable option "nodosfilewarning" turns off this warning.
Consult the user's guide for more details about POSIX paths:
http://cygwin.com/cygwin-ug-net/using.html#using-pathnames

TCPserver Waiting for client on port 5000

Got a connection from (169.254.237.97 , 53462)
SEND file to USIM??(q or Q to quit) :

The Mac code generated--
b7      2a      a9      b7

The Encrypted Message sent to the USIM is

j      p      q      1      ♦      ?
      +      ~      I      <      I
      D
3      k      8      C      I
f      <      X

Waiting for response from USIM--
```

**FIG10: SERVER (RNC) SIDE**

The above screenshot is taken on server machine. The server creates a socket on port 5000 and publishes its IP. It then waits for connection from any client. As can be seen from fig it gets a connection from client with IP 169.254.237.97 and port 53462. The server first computes the MAC code on the input message using the function f9 and integrity key IK. It then encrypts the input file using function f8 and cipher key CK. It then appends the MAC code to the end of the file and sends it to the client (or USIM). It then waits for response from the client.

```

C:\cygwin\bin\sh.exe
cygwin warning:
MS-DOS style path detected: C:\Program Files\NetBeans 6.7.1\dlight1\bin\native
execution\dorun.sh
Preferred POSIX equivalent is: /cygdrive/c/Program Files/NetBeans 6.7.1/dlight
1/bin/nativeexecution/dorun.sh
CYGWIN environment variable option "nodosfilewarning" turns off this warning.
Consult the user's guide for more details about POSIX paths:
http://cygwin.com/cygwin-ug-net/using.html#using-pathnames

The encrypted message received from RNC --
D~k8 3I C J f < X * ? 1 . < I ? N
b7 2a a9 b7

The received MAC on USIM---
ffffffb7 2a fffffffa9 fffffffb7

The calculated MAC on USIM---
ffffffb7 2a fffffffa9 fffffffb7

Checking calculated MAC with received MAC--
Successfully matched!!

Integrity of incoming message from RNC verified!!

The decrypted message from RNC is
H i t h i s
i s r v e s h e l s e t m
e s t h s e t a r o t i
t s t h s e t a r o t i
m u n i c a t i
o n

Enter the message to be sent to the RNC--
SEND <q or Q to quit> :

```

**FIG11: CLIENT (USIM) SIDE**

Here the screenshot is taken on the client machine. It receives the encrypted file from the server (or RNC) and extracts the MAC code. It then computes the MAC code on its own machine using the function f9 and integrity key IK. It then checks the computed MAC with the received MAC. If they match, then the integrity of the received message is verified. The USIM then decrypts the message using the function f8 and cipher key CK and writes the result in a local file.

---

## *CHAPTER 5 CONCLUSION AND FUTURE WORK*

---

### 5.1) CONCLUSION

In this thesis we outlined the 3G Rel99 architecture and the framework of the 3G security architecture. We have discussed the main features of 3G security architecture and its improvements over the 2G GSM system. Security mechanisms like two way authentication, integrity protection of signaling data and the extension of security to the core network are robust and can successfully prevent most of the threats and intrusion from potential hackers. However there are a few loopholes like transmission in the open of permanent user identity in the initial allocation of temporary identity and user domain data not integrity protected, that may be exploited by potential hijackers.

### 5.2) FUTURE WORK

In this thesis we have implemented the security algorithms to protect the interface between the mobile station and the RNC (network access security). This implementation can be extended to the security features like MAPSEC[9] and IPSEC[10] for protection of the core network(network domain security).

## REFERENCES

- [1] C. Xenakis, L. Merakos, "Security in third Generation Mobile Networks", *Computer Communications*, Vol.27, pp. 638-650, 2004.
- [2] "Evaluation of UMTS security architecture and services", A. Bais, W. Penzhorn, P. Palensky, Proceedings of the 4th IEEE International Conference on Industrial Informatics, p. 6, Singapore, 2006.
- [3] UMTS security, Boman, K. Horn, G. Howard, P. Niemi, V. Electronics & Communication Engineering Journal, Oct 2002, Volume: 14, Issue: 5, pp. 191-204.
- [4] Colin Blanchard, "Security for Third Generation (3G) Mobile Systems" Elsevier Science, Information Security Technical Report, Vol.5, No. 3, 2000.
- [5] 3GPP TS 21.133 (v3.2.0), 3G Security, Security Threats and Requirements, Release '99, Dec 2001.
- [6] 3GPP TS 23.002 (v3.5.0), Network Architecture, Release '99, Sep 2002.
- [7] 3GPP TS 33.102(v3.12.0), 3G Security, Security Architecture, Release '99, June 2002.
- [8] 3GPP TS 33.900(v1.2.0), A Guide to 3G Security, Jan 2000.
- [9] 3GPP TS 33.200(v4.3.0), 3G Security, Network Domain Security; MAP application layer security, Release 4, March 2002.
- [10] 3GPP TS 33.210(v5.1.0), 3G Security; Network Domain Security; IP application layer security, Release 5, June 2002.
- [11] 3GPP TS 33.800 "3G Security, Principles for Network Domain Security", Release 4/5, Oct 2000.

[12] 3GPP TS 35.201(v3.2.0), 3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 1: f8 and f9 Specification, Release '99, Dec 2001.

[13] 3GPP TS 35.202(v3.1.2), 3G Security, Specification of the 3GPP Confidentiality and Integrity Algorithms, Document 2: KASUMI Specification, Release '99, Aug 2001.

[14] 3GPP TS 35.205(v3.0.0), 3G Security, Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*, Document 1: General, Release '99, Apr 2001.

[15] 3GPP TS 35.206(v3.0.0), 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1\*, f2, f3, f4, f5 and f5\*; Document 2: Algorithm Specification, Release '99, Apr 2001.